Circuit and CAD Techniques for Expanding the SRAM Design Space

James Boley
Department of Electrical and Computer Engineering
University of Virginia

A Dissertation Proposal Presented in Partial Fulfillment of the Requirement for the
Doctor of Philosophy Degree in Electrical Engineering
April 24, 2013

Abstract

As mobile devices become heavily energy constrained, the need for low power, energy efficient circuits has emerged. The application space varies from ultra low power devices such as body sensor networks (BSNs), to higher performance applications such as smart phones, tablets, and all other devices constrained by battery life. In order to reduce energy consumption and increase energy efficiency, voltage supplies are scaled down to take advantage of quadratic active energy savings. Static random access memory (SRAM) is a critical component in modern system on chips (SoCs), consuming large amounts of area and often on the critical timing path. SRAM is the most commonly used in cache designs due to its high speed and high density. In the past, conventional SRAM designs were able to take advantage of Moore's law by simply reducing devices sizes and scaling down V_{DD}. This has become increasingly difficult as devices enter the nanoscale range due to increased device variability and leakage. SRAM devices are typically minimum sized, which further compounds this problem. The increase in both variation and leakage leads to reduced read and write margins, making it more difficult to design low power SRAMs that meet frequency and yield constraints. In addition, as the capacity of SRAM arrays continues to increase, the stability of the worst case bitcell degrades. Therefore it has become increasingly important to evaluate the effect of V_{DD} reduction on SRAM yield and performance.

The goal of this work is to push the memory design space beyond its conventional bounds. Typically the minimum supply voltage (V_{MIN}) of SRAMs is higher than that of conventional CMOS logic due to a higher sensitivity to device variation. In order to push SRAM designs past this apparent brick wall, new knobs have been introduced such as alternative bitcells and read and write assist methods which improve the robustness of SRAMs in the presence of variability. These knobs introduce new tradeoffs between energy, speed, area and yield which are difficult to evaluate because they are dependent on many factors such as technology node, bitcell architecture, and design constraints.

In this work, we first investigate the tradeoffs in designing a subthreshold SRAM embedded in an ultra low power body sensor network. The result of this work is one of the first embedded subthreshold memories, capable of operation down to 0.3 volts. Next, we present a method for fast, accurate estimation of SRAM dynamic write V_{MIN}, which we will show provides a speedup of 112X over statistical blockade at a cost of only 3% average error. Furthermore, we will evaluate the combination of new bitcell circuit topologies and circuit assist methods at reducing SRAM read and write V_{MIN}. Next, we extend the functionality of an existing tool used for rapid design space exploration and optimization of SRAMs. The proposed extensions include: evaluation of read and write assist methods, support of multi-bank design evaluation, circuit and architectural level co-optimization engine, and yield evaluation. Finally, we propose a method for tracking PVT variations during the write operation in order to regain energy lost through over-conservative guard-banding. The anticipated contribution of this research is a set of methods and tools for pushing SRAM designs to lower operating voltages, increasing yields, and evaluating design tradeoffs.

1 Introduction

1.1 Motivation for Reducing SRAM V_{MIN}

As mobile devices become heavily energy constrained, the need for low power, energy efficient circuits has emerged. In order to reduce energy consumption and increase energy efficiency, voltage supplies are scaled down to take advantage of quadratic active energy savings. Static random access memory (SRAM) is a critical component in modern system on chips (SoCs); consuming large amounts of area and often on the critical timing path. SRAM is the most commonly used in cache designs due to its high speed and high density. In the past, the voltage of these memories has been easily scaled down with technology; however recent increases in variability and leakage have presented new design challenges. The increase in both variation and leakage leads to reduced read and write margins, making it more difficult to reduce the minimum operating voltage (V_{MIN}) of SRAM designs. This problem is compounded by the fact that SRAMs typically use minimum sized to reduce area [1]. In addition, as the capacity of SRAM arrays continues to increase, the stability of the worst case bitcell degrades. Therefore it has become increasingly important to accurately evaluate the effect of V_{DD} reduction on SRAM yield and performance.

In addition to reducing active energy, reducing V_{DD} also reduces leakage energy. This is especially important for SRAMs due to the fact that memories can contain millions of cells and can consume up to 90% of the total chip area. Therefore a small reduction in the leakage energy per cell, results in a significant overall energy saving.

1.2 Key Challenges in Reducing SRAM V_{MIN}

1.2.1 Reduced Read Static Noise Margin

The static noise margin is typically calculated using the butterfly curve technique (Figure 1) first introduced by [2]. This metric is a measure of the amount of noise that a bitcell can tolerate before its data becomes corrupted. During a read operation, both of the bitlines are precharged high, and are held dynamically at V_{DD}. Once the wordline (WL) pulses high, the charge stored on the BL is discharged through XL and NL (Figure 1). Because the bitline is shared with many cells (up to 512), the value of C_{BIT} is very large. This can cause the node at Q to rise above ground. In order to ensure that the voltage at this node does not rise above the switching threshold of the PR/NR inverter, the resistance of the XL transistor must be kept larger than that of the NL transistor. If the voltage rises above the threshold value

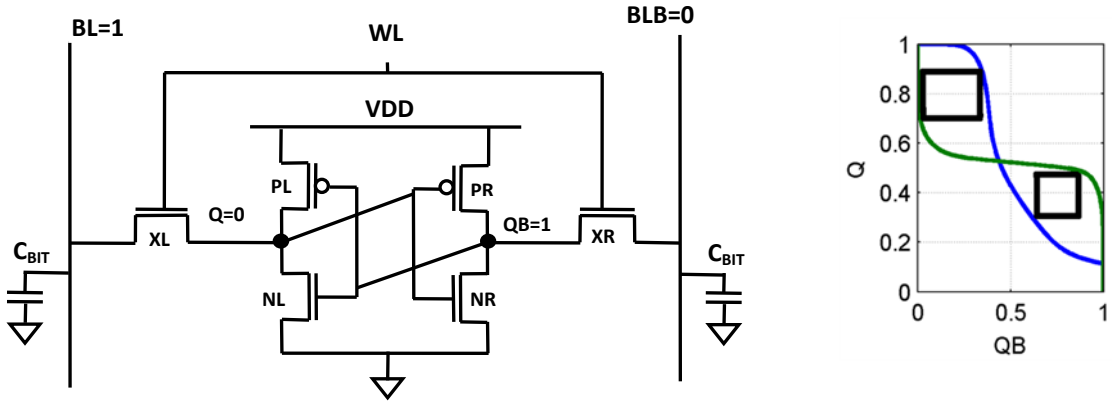

Figure 1. The 6T bitcell schematic (left) and the methodology for measuring static noise margin (right)

of NR, this could result in the data being stored to flip values. This is prevented by sizing the pull-down and passgate according to equations 1-3.

$$k_{n,XL}\left[(V_{DD}-\Delta V-V_{Tn})V_{DSATn}-\frac{V_{DSATn}^2}{2}\right]=k_{n,M1}(V_{DD}-V_{Tn})\Delta V-\frac{\Delta V^2}{2} \quad (1)$$

$$\Delta V=\frac{V_{DSATn}+CR(V_{DD}-V_{Tn})-\sqrt{V_{DSATn}^2(1+CR)+CR^2(V_{DD}-V_{Tn})^2}}{CR} \quad (2)$$

$$CR=\frac{W_{NL}/L_{NL}}{W_{XL}/L_{XL}} \quad (3)$$

As an example, if the threshold voltage of the NMOS transistor is 0.4 volts, than the cell ratio (CR) must be kept above 1.2 in order to ensure that the voltage of the Q node (ΔV) does not rise high enough to turn on the NR transistor. By sizing these devices properly, we can ensure that the bitcell remains stable during a read. However, as we can see from these equations, variation in threshold voltage could cause the bitcell to become unstable. This type of ratioed design becomes even more unreliable in subthreshold where the on current becomes exponentially dependent on V_T (equation 4).

$$I_D=I_O\frac{W}{L}exp\left(\frac{V_{GS}-V_{th}-\eta V_{ds}}{nV_T}\right)\left(1-exp\left(-\frac{V_{ds}}{V_T}\right)\right) \quad (4)$$

1.2.2 Reduced Write-Ability

During a write (Figure 2a), the bitlines are driven statically to V_{DD} and ground. In this example we are writing a '1' into the cell. Because we have sized the XL/NL ratio such that the Q node cannot rise high enough to flip the cell, the new value must be written in by pulling the QB node to ground. Again in this case we have a ratioed fight occurring, this time between the XR and PR transistors. In order to write a '0' into the bitcell, the QB node must be pulled low enough to turn on the PL transistor. Using a similar approach as in section 1.2.1, we can set the currents of these two transistors equal in order to

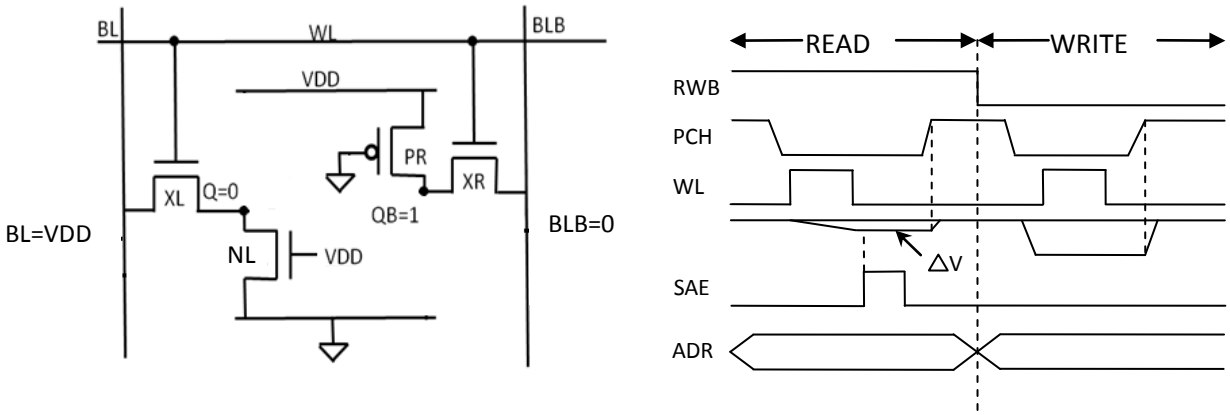

Figure 2. The 6T bitcell schematic during a write (a), and a typical timing diagram for an SRAM (b)

determine the minimum sizing of the pull up to pull down ratio. What we find is that the pull up device should typically be kept minimum sized in order to improve write-ability. The downside to this is that the variability of this device will be larger due to the fact that it is minimum sized. As with read-stability, write-ability is reduced in subthreshold due to the exponential dependence of the on current to threshold voltage variations.

1.2.3 Read Access Fails

Read access fails occur when the bitline differential developed before the sense amp enable (SAE) signal goes high is not large enough for the sense amp to correctly resolve to the correct value (Figure 2b). This occurs due to variation in both the maximum current being sunk by the bitcell during a read (I_{READ}), and the sense amp offset voltage due to variation within the sense amp (V_{OS}). I_{READ} sets the delay for the proper BL differential to develop and is typically normally distributed. V_{OS} determines the minimum BL differential required in order for the sense amp to resolve to the proper value. The sense amp offset is also normally distributed and typically has an average of 0 mV. A read access failure is usually considered a performance failure, because the read failed to complete within the cycle time. It has been shown in [3] that 55% of the total read delay occurs in the development of the BL differential. Therefore it is important to minimize the delay between the WL and SAE signal ($T_{WL\text{-}SAE}$) without compromising yield. Worst case analysis sets the value of $T_{WL\text{-}SAE}$ by pairing the worst case bitcell

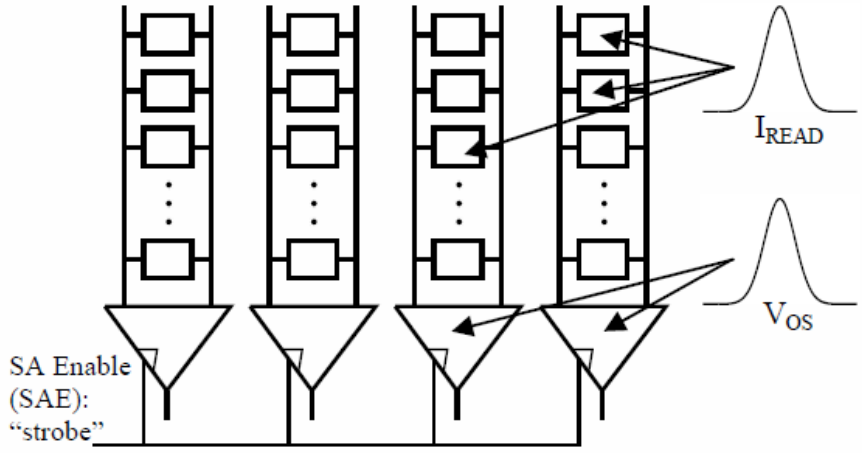

Figure 3. Read access fails occur due to variation in read current and built-in sense amp offset [3]

with the worst case sense amp. However it is noted in [3] that the probability of this occurring in a large memory is actually very small. By using this pessimistic approximation, we are sacrificing performance as well as energy. The increase in energy is due to the fact that the WL pulse width is larger than it needs to be, resulting in more charge being dissipated from the bitlines. [3] instead uses order statistics to determine the bitcell/sense amp pairing that results in the worst case $T_{WL\text{-}SAE}$.

1.2.4 Estimating Yield

Monte Carlo (MC) simulation is the gold standard for evaluating the effects of variation on circuit performance and reliability. Because variation is a stochastic process, we use MC to calculate failure probabilities, but can't necessarily guarantee functionality. The difficulty with using MC for SRAMs is that memories can contain millions of bits, causing the number of simulations needed for margining to become prohibitively large. In addition, because we are only concerned about points lying in the tail

region, Monte Carlo simulations are not efficient at identifying these points. Therefore, we need some method for quickly and accurately estimating SRAM failure probabilities.

1.2.5 Evaluating Design Decisions

The introduction of new circuit techniques such as read and write assist methods and new bitcell topologies creates a whole new set of tradeoffs between speed, area, performance and reliability. These tradeoffs are difficult to evaluate because they are dependent on many factors such as technology node, bitcell architecture, and design constraints. Therefore, a change in any one of the key memory circuits or in the core cell technology will alter the optimal circuit topologies, partitioning, and architecture for the entire memory. We can no longer innovate in one portion of the memory while ignoring the effects our innovation could have on the overall memory and system design. Without the proper support structure and tools, it would be nearly impossible to re-design and re-optimize an entire memory by hand every time we try a new circuit, much less explore a technique's impact across different technologies and applications.

1.3 Goals

The goal of this work is to push the memory design space beyond its conventional bounds. Typically the V_{MIN} of SRAMs is higher than that of conventional CMOS logic due to a higher sensitivity to device variation. In this work, we will focus on developing methods and tools to push SRAM designs past this apparent brick wall. The major goals are as stated:

- Propose a methodology for designing reliable, embedded sub-threshold SRAM. Designs decisions such as choice of bitcell topology, use of read and write assist methods, architectural topologies and timing will be evaluated in terms of system requirements such as energy and timing constraints.
- Evaluate the effect of four read and two write assist methods on yield and V_{MIN} reduction.
- Propose a methodology for quickly evaluating dynamic write V_{MIN} through simulation. The methodology will be evaluated in terms of speed up over existing techniques as well as accuracy.
- Extend the existing Virtual Prototyping (ViPro) tool to perform optimization of multi-bank caches. The importance of supporting muli-bank memories is that it will allow the tool to optimize across a larger range of memory capacities, thus increasing the optimization design space.
- Extend ViPro to support optimization using the 8T bitcell. This cell is commonly used in level one caches due to its dual port design and creates new design challenges due to its single ended read structure.
- Support optimization using 3 different read and write assist methods. The goal of this work is to evaluate the effect of each method on speed, energy, and yield.
- Support design optimization using an optimization engine. The optimizer will be evaluated in terms of speed-up over brute force optimization.
- Propose an adaptive method for minimizing SRAM write V_{MIN} by monitoring PVT variation on chip. This method will be evaluated by the total energy savings gained over traditional guard-banding techniques without sacrificing yields.

Thesis Statement: Reducing SRAM V_{MIN} in order to improve energy efficiency is one of the major challenges facing memory designers today. Voltage scaling in modern SRAM designs has become increasingly difficult due to increased variability and leakage, leading to reduced reliability. The anticipated contribution of this research is a set of methods and tools for pushing SRAM designs to lower operating voltages, increasing yields, and evaluating design tradeoffs.

2. Subthreshold SRAM Design for a Body Area Sensor Node

2.1 Motivation

Body sensor nodes (BSNs) promise to provide significant benefits to the healthcare domain by enabling continuous monitoring and logging of patient bio-signal data, which can help medical personnel to diagnose, prevent, and respond to various illnesses such as diabetes, asthma, and heart attacks [4]. One of the greatest challenges in designing BSNs is supplying the node with sufficient energy over a long lifetime. A large battery increases the form factor of the node, making it unwearable or uncomfortable, while a small battery requires frequent changing and reduces wearer compliance. Another option is to use energy harvesting from ambient energy sources, such as thermal gradients or mechanical vibrations in order to provide potentially indefinite lifetime [4]. However, designing a node to operate solely on harvested energy requires ultra-low power (ULP) operation since the typical output of an energy harvester is in the 10's of μWs [5]. To ensure sustained operation of the node using harvest energy, on-node processing to reduce the amount of data transmitted, power management, and ultra-low power circuits are critical.

In order to achieve ULP operation, voltages must be scaled down to reduce both active and leakage energy. The sub-threshold region ($V_{DD}<V_T$) has been shown by [6] to minimize energy per operation. Sub-threshold systems require SRAMs for storing data at these low voltages. The problem is that while logic has been shown to easily scale into the sub-threshold region, the traditional 6T SRAM bitcell becomes unreliable at voltages below 700 mV due to process variations and decreased device drive strength [7]. SRAM devices are typically minimum sized which further compounds this problem. Therefore, in order to design reliable SRAMs, capable of operating in the sub-threshold regime, more robust bitcell designs must be used.

2.2 Prior Art

The 8T bitcell [8] shown in Figure 4a adds a two transistor read buffer to the conventional 6T bitcell in order to prevent the data from being disturbed during a read. In a normal read operation, the bitlines are precharged and the WL is pulsed high, causing the bitcell to discharge one of the bitlines. The problem with this is that if the node storing a '0' rises above the switching threshold of right inverter (Figure 4a), then the cell could unintentionally flip. The 8T cell solves this problem by decoupling the data from the read operation; therefore the read SNM becomes the hold SNM. One weakness of this bitcell is that it still suffers from half-select instability, which occurs during a write when an unselected cell is read like a traditional 6T bitcell. Currently the best method to solve this problem in a bit interleaved architecture is by using a read before write scheme. In this method the entire row is read and then the data is written back into the unselected cells at the same time that new data is written to the selected cells.

The 10T bitcell [9] (Figure 4b) uses Schmitt Trigger (ST) inverters to help improve the read static noise margin (RSNM). The NR2/NFR feedback transistors weaken the pull down network when VR is high, increasing the switching threshold of the right inverter. This means that the VL node would have to pull up much higher during a read in order to flip the cell, resulting in higher read stability. This bitcell has been shown by [9] to have 1.56X higher read SNM compared to the conventional 6T bitcell. The downside to this topology is that the four extra transistors result in a 33% area penalty compared to the 6T bitcell.

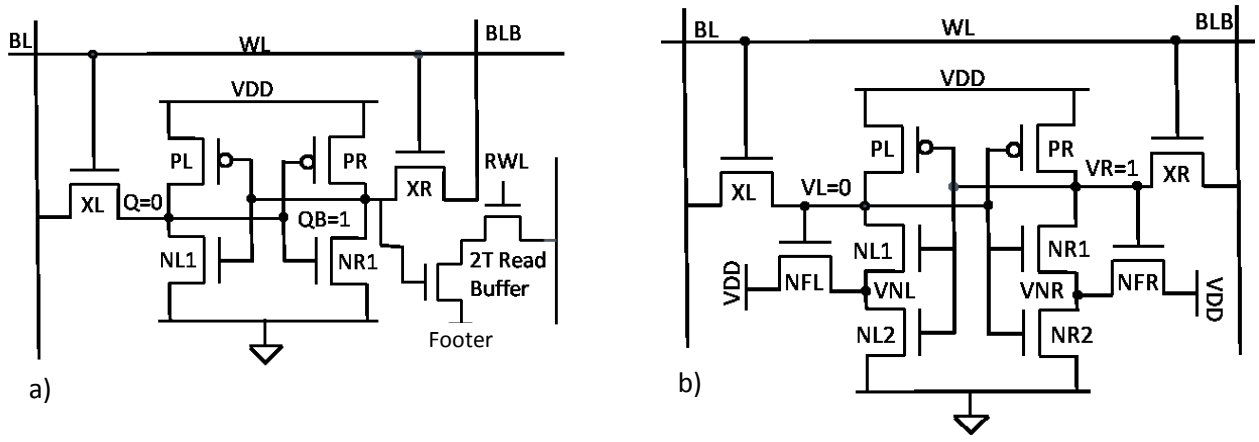

Figure 4. (a) 8T bitcell with read buffer [8]; (b) 10T Schmitt Trigger inverter bitcell [9]

2.3 Research Question

How can we design an embedded SRAM capable of reliable operation at 500 mV which will meet the timing constraints of the system?

2.4 Approach

The first version of the BSN chip required a 1.5 kB instruction SRAM / ROM and 4kB data SRAM. The instruction memory (IMEM) was required for storing 12 bit instructions for execution by the digital power management (DPM) block and the PIC processor. It is programmed once during startup using a scan chain, then once the chip is deployed, the memory is only used for reading out instructions. The data memory (DMEM) is used as a FIFO (First In, First Out). During signal acquisition, the digital data is streamed directly into the DMEM. Once the memory is full, the memory address is reset to 0 and old data is replaced with new data. When an atrial fibrillation (Afib) event is detected, the previous eight heart beat samples stored in the data memory are transmitted wirelessly from the radio.

The first step in the design process was designing a reliable bitcell. The three metrics that we considered were: read static noise margin, write noise margin and read access stability. Monte Carlo simulation showed that the mean-3σ point (for RSNM) was around 15 mV (Figure 5a). With a margin this low, any noise source on the supply could potentially result in an accidental bit flip during a read. Therefore to remedy this issue we decided to use the 8T bitcell, which as described in section 2.2, eliminates the problem of read instability in designs that do not use bit interleaving. In order to eliminate the half-select instability that occurs during a write, a row buffer is used to store the eight words per row. A write only occurs when the row buffer is full and the entire row is then written. Since each row of the DMEM contains eight 16-bit words, the memory is only written once every eight cycles. This control is managed by the DMA which is a subthreshold accelerator to interface the DMEM with the rest of the SoC. We are able to use this approach due to the fact that the DMEM is used as a FIFO (First-in, first-out), where each successive write increments the word address by one. This same technique is used to write the IMEM, however the control in this case is through the use of a scan chain. During a read, both the instruction and data memories output the entire row, and the individual word is selected by the DPM (IMEM) or the DMA (DMEM). This type of design allows us to reduce the number of reads and write to once every eight cycles, thus achieving close to an 8x energy savings (minus the overhead of additional buffers).

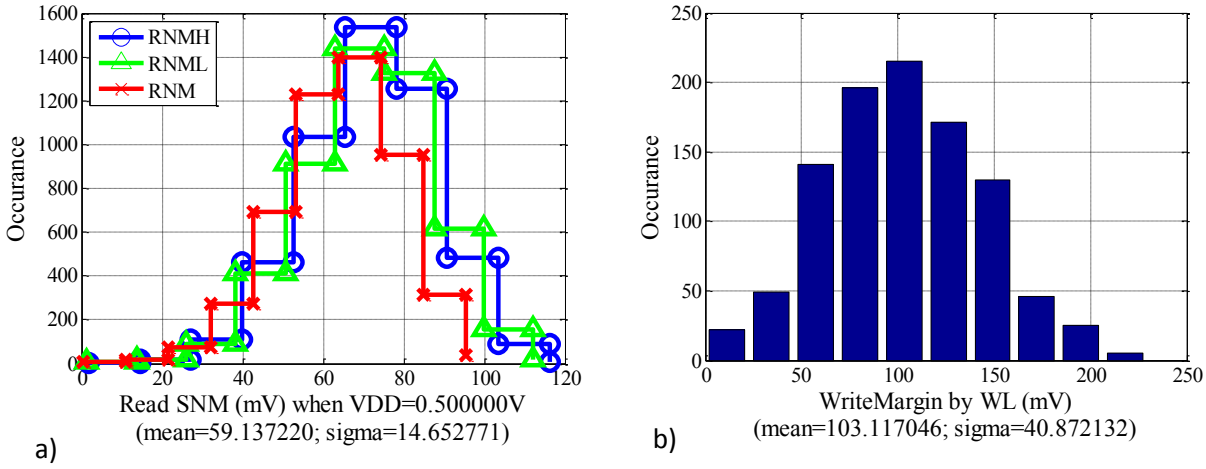

Figure 5. a) Read SNM distribution at 500 mV b) write margin distribution at 500 mV

The next metric to consider is write noise margin. Because leakage is a major concern in SRAMs due to the large number of inactive bitcells, the ideal bitcell would use high V_T devices to reduce this wasted energy. However, through Monte Carlo analysis we found that the worst case static noise margin of the bitcell using high VT devices was close to zero, meaning the bitcells were failing to write (Figure 5b). Therefore in order to ensure adequate write margins, we decided to use regular V_T devices. Using these devices, we were able to achieve a worst case write margin of 100 mV. The downside to using regular V_T devices is that it increases the leakage current per bitcell by 24X.

The final metric to consider to ensure reliability is read access stability. Typically in super-threshold, read stability is determined by the minimum BL differential required for the sense amp to generate the proper output. However because speed is not an issue due to the 5 microsecond cycle time, no sense amp is required. Because the 8T bitcell has single ended reads, the output of the RBL is fed directly into a standard buffer. The real concern for this memory is that the leakage current from the unaccessed cells does not cause the RBL to droop when a '1' is being read. By reducing the number of bitcells per column, we can reduce the total leakage current, however this results in a larger number of banks. Having more banks increases the total area due to increased redundancy of the periphery cells (WL drivers, BL drivers, output buffers). Another approach is to reduce the leakage from the unaccessed rows by precharging the footer voltage (Figure 4a) to V_{DD}. [8] shows that this technique reduces the RBL leakage to almost zero. This technique does however introduce a new problem. Because the footer must be driven to V_{DD}, when a row is active, the footer must sink all of the current from each column (in this design there are 128 columns). By using a charge pump to boost the input gate voltage of this buffer to $2*V_{DD}$, we are able to achieve a ~13.5X increase in on current. Even with this increase in current, we found that the maximum number of bitcells per column to ensure that the RBL pulled low within a single cycle was 64.

In addition, the DMEM was split into four 1 kB banks that can be individually power gated by NMOS footers being overdrive to 1.2V when active to ensure low levels of ground bounce. Overdriving the gate to 1.2V allowed for smaller footer widths, resulting in reduced leakage current in sleep mode. We chose to use NMOS footers because the N-P ratio (ratio of the NMOS on current, to PMOS on current) was ~10. This meant that the PMOS switched had to be upsized by a factor of 10X to achieve the same amount of on current.

2.5 Evaluation Metrics

The design will be evaluated on two metrics: minimum operating voltage at which reliable operation is achievable and total energy per access. Success is defined as reliable operation down to at least 0.5 volts at 200 kHz (operating voltage and frequency of the system).

2.6 Results and Contributions

The design was fabricated in a 130nm commercial process. The data and instruction memories were designed fully custom using Cadence. Results show reliable operation down to 0.3V at 200 kHz. IMEM read energy was measured at 12.1 pJ per read at 0.5V and leakage energy per cycle of 6.6 pJ. To our knowledge, this memory is the first embedded 8T SRAM capable of operating in subthreshold without the use of assist methods.

2.7 Future Work

Because we chose to use standard V_T devices to ensure write-ability, the leakage energy was relatively high compared to the total energy of the chip. To reduce this leakage energy, the bitcell should be designed with high V_T devices. However to ensure reliable operation, read and write assist methods will likely need to be implemented.

3. A Method for Fast, Accurate Estimation of SRAM Dynamic Write V_{MIN}

3.1 Motivation

Because SRAM memories can contain millions of cells, it is important to accurately predict the reliability of the worst case bitcell in order ensure reliability. The most common method for evaluating yield is through Monte Carlo (MC) simulations. However for very large arrays (i.e. 10 Mb) the number of simulations required to identify the worst case bitcell becomes prohibitively large. Because the majority of simulated samples do no lie in the tail region, a full MC simulation is not an efficient method for estimating very small failure probabilities. A common approach to reducing simulation time is to run a relatively small number of samples and then fit the resulting distribution to the normal distribution. Once the μ and σ are known, the stability of the worst case bitcell can be identified. The problem with this approach is that it can only be applied to data sets that replicate a known distribution [10][11]. However, it has been shown that the dynamic write margin does not fit the normal distribution [11][12]. The distribution resembles the long tail F-distribution, but does not match it exactly. Because the distribution does not closely match any known statistical distribution, it is difficult to model without full simulation of the tail region.

3.2 Background

The dynamic noise margin is defined as the minimum pulse width required to write the cell, or T_{CRIT} [12-18]. The benefit of this metric is that it takes into account the transient behavior of the bitcell, which is not captured by static metrics. This metric has been shown by [16] to produce more accurate V_{MIN} estimations than static metrics, since static metrics give optimistic write margins and pessimistic read margins, due to the infinite wordline (WL) pulse width. In this paper we focus primarily on dynamic write-ability since the static metric results in optimistic yields and because it has been shown that write failure is more likely in newer technologies [19]. The downside to using transient simulations is that they are more time costly, especially when running large numbers of Monte Carlo samples to isolate the worst

case bitcells. Whereas as static margin can be calculated using a single simulation, the calculation of T_{CRIT} requires a binary search. This takes on average ten to fifteen iterations to accurately determine the critical pulse width with a high level of accuracy.

3.3 Prior Art

One approach to solve this problem is to develop purely analytical models as in [20][21]. However these approaches are less accurate because approximations must be made to simplify the problem. [12] showed that these approximations can lead to errors in failure probability estimates of up to three orders of magnitude. Two methods that reduce MC run time by effectively simulating only points in tail region include importance sampling [22][23] and statistical blockade [24][25]. These techniques can be used to reduce simulation time by several orders of magnitude. However, in order to accurately determine the dynamic margin using binary search, its takes an average of twelve simulations. Using this method, it would take over 894,000 simulations to identify the worst case write margin for a 100 Mb memory.

In [10][11] the author defines static V_{MIN} under the presence of variation. The V_{MIN} is defined as the point where the SNM becomes zero. The author uses the hold SNM to define the data retention voltage, the read SNM to define read V_{MIN}, and the WL sweep method to define write V_{MIN} [26]. To estimate the failure probability at a given supply voltage, each metric is simulated across a range of VDDs. Each resulting distribution is then fitted to the normal distribution. As V_{DD} is reduced, the mean of the write distribution decreases and the standard deviation increases. Then using equations (4) and (5), the failure probability can be calculated for any V_{DD}. In equation (1), s is equal to the SNM which causes a failure, which in this case is just zero. μl and μh are defined as the SNM for writing a zero and writing a one. Equation (5) is a best fit line representing the value of μ and σ versus V_{DD}.

$$p_f = \frac{1}{2}erfc\left(\frac{\mu_{h-s}}{\sqrt{2}\sigma_0}\right) + \frac{1}{2}erfc\left(\frac{\mu_{l-s}}{\sqrt{2}\sigma_1}\right) - \frac{1}{4}erfc\left(\frac{\mu_{h-s}}{\sqrt{2}\sigma_0}\right) * \frac{1}{4}erfc\left(\frac{\mu_{l-s}}{\sqrt{2}\sigma_1}\right) \quad (4)$$

$$\mu = \mu_0 + a(v^2 - v_0^2) + b(v - v_0), \quad \sigma = \sigma_0 + c(v - v_0) \quad (5)$$

The problem with this approach is that the dynamic margin is not normally distributed. From Figure 6,

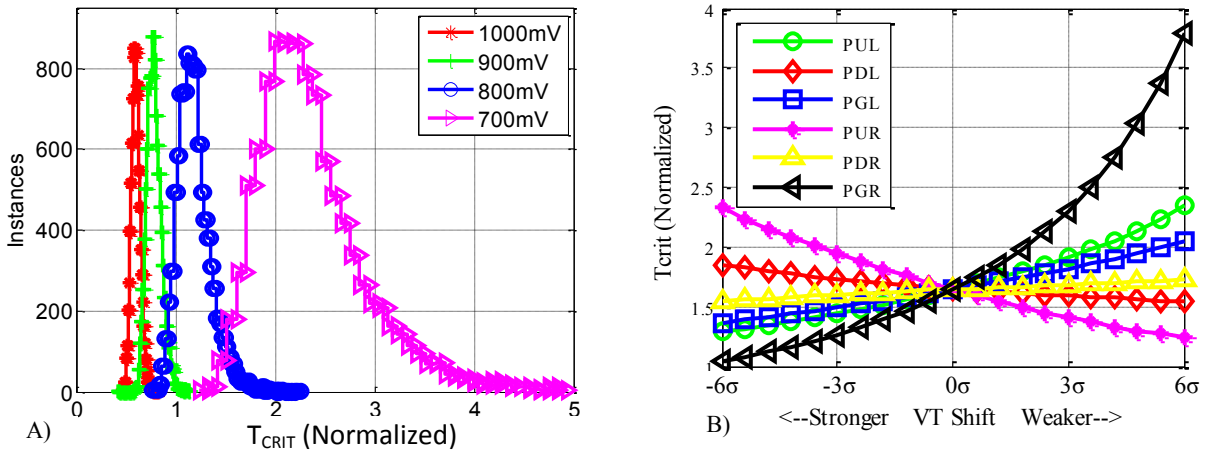

Figure 6. A) The distribution of T_{CRIT} does not fit a normal distribution, B) In order to characterize the bitcell, the VT of each transistor is swept independently

the shape of the T_{CRIT} distribution is long tailed, making the normal approximation inaccurate. Therefore a new methodology must be created to accurately predict the tail of the dynamic margin distribution.

3.4 Hypothesis

We hypothesize that by using sensitivity analysis we can further reduce the time required to calculate dynamic write V_{MIN} with only a small accuracy penalty.

3.5 Approach

In order to reduce the cost of running large numbers of transient Monte Carlo simulations, we propose using sensitivity analysis to quickly generate the T_{CRIT} distribution [27]. The first step in this method is to sweep the threshold voltages of each transistor to produce the plot shown in Figure 6. The PU, PD, and PG labels represent the pull-up, pull-down, and passgate transistors respectively. The left node of the bitcell is initially holding a '0' and the right node is initially holding a '1'. The x-axis represents the V_T shift of each transistor ranging from -6σ to 6σ; the y-axis represents the resulting T_{CRIT} value. When sweeping the V_T of each transistor, all other transistors are left at nominal V_T. We then fit each curve to a third order polynomial:

$$T_{\text{CRIT-OFFSET}} = aV_{T-Shift}^3 + bV_{T-Shift}^2 + cV_{T-Shift} + d \quad (6)$$

Once each of the curves has been fitted, the next step is to generate a V_T distribution for each of the six transistors (Figure 7). This is done by generating a normal distribution using the sigma values from the Spice model. Next, the V_T offset of each transistor is plugged into (6), and the six offsets are then added to the nominal case to produce the T_{CRIT} prediction:

$$T_{\text{CRIT}} = T_{\text{CRIT-NOM}} + T_{\text{CRIT-OFFSET-PUL}} + \ldots + T_{\text{CRIT-OFFSET-PGR}} \quad (7)$$

This calculation is repeated N times depending on the desired sample size. Clearly, computing (7) is much faster than running the set of simulations required to find T_{CRIT} using Spice.

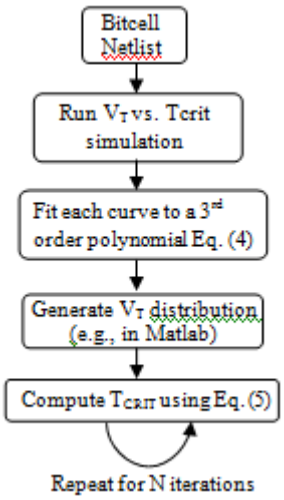

Figure 7. Flowchart of the proposed T_{CRIT} estimation methodology

3.6 Evaluation Metrics

This method will be evaluated on two metrics: speedup gained over existing methods and loss of accuracy. A successful method will maximize speedup and minimize loss of accuracy.

3.7 Contributions

In order to verify the accuracy of this methodology, we compared the margin of the worst case bitcell calculated by the model and using the recursive statistical blockade tool [25]. The accuracy of the model was tested for three memory sizes: 100 Kb, 10 Mb, and 100 Mb. The model was also tested across a range of VDDs from 500 mV up to 1V. The results are shown in Table 1. We can see from the table that the worst case error is only 6.83%, while the average is 3.01%. A positive percentage error means that the model overestimated the T_{CRIT} value, resulting in slightly pessimistic margins.

The advantage of this method is that it greatly reduces simulation times while sacrificing very little accuracy compared to statistical blockade. This same technique can be applied to importance sampling to reduce the total run time. Simulating the VT curves in Figure 4 requires approximately 18.8 minutes. Once these curves have been produced, random samples are generated (e.g., by MATLAB) and applied to (5). The run time for the sensitivity analysis increases linearly with the number of samples. The total run time for a 100 Mb memory is only 32 minutes. One disadvantage of the statistical blockade tool is that in order to determine the worst case write margin, two separate test cases must be run: writing a '0' and writing a '1'. This means that two separate filters must be generated, as well as two separate sets of Monte Carlo simulations. The total number of simulations required for the recursive statistical blockade tool is 894,288, corresponding to a total CPU runtime of 60 hours.

Table 1. a) The percentage error of the sensitivity analysis versus statistical blockade for varying memory sizes accross V_{DD}. b) A comparison of the run times between statistical blockade and sensitivity analysis

Modeled data vs. Statistical Blockade (Percentage Error)			
	100K	10M	100M
500 mV	6.83	-4.25	6.51
600 mV	2.96	-3.69	5.61
700 mV	-0.18	-2.64	4.75
800 mV	0.83	-0.7	1.21
900 mV	-4.5	0.83	1.43
1000 mV	-2.72	-2.2	-2.27
Average	3.01	2.39	3.63

	Statistical Blockade	Sensitivity Analysis
	Num. simulations	Run Time
Initial Simulation	24,000	18.8 min
100 Kb	107,904	0.72 s
10M	531,096	72 s
100M	231,288	12 min
Total Simulations	894,288	
Total Run Time	60 Hours	32 minutes

In summary, our method provides a 112.5X speedup at the cost of an average loss in accuracy of 3.01% and a worst case loss of 6.83%

4. Analyzing Sub-threshold Bitcell Topologies and the Effects of Assist Methods on SRAM V_{MIN}

4.1 Motivation

As mobile devices become heavily energy constrained, the need for ultra low power circuits has emerged. In order to reduce energy consumption, voltage supplies are scaled down to take advantage of

quadratic energy savings. The sub-threshold region ($V_{DD}<V_T$) has been shown by [6] to minimize energy per operation. Sub-threshold systems require Static Random Access Memory (SRAM) for storing data at these low voltages. The problem is that while logic has been shown to easily scale into the sub-threshold region, the traditional 6T SRAM bitcell becomes unreliable at voltages below 700 mV due to process variations and decreased device drive strength [7]. SRAM devices are typically minimum sized, which further compounds this problem. As the capacity of SRAM arrays continues to increase, the stability (typically measured in terms of Static Noise Margin (SNM) [2]) of the worst case bitcell degrades. Therefore, in order for the minimum operating voltage (V_{MIN}) of SRAMs to enter the sub-threshold regime, more robust bitcell designs or assist methods must be used.

4.2 Hypothesis

Using different combinations of bitcell topologies and assist methods, we can determine which approach results in the largest reduction of read and write V_{MIN} over the nominal case.

4.3 Approach

4.3.1 Write Assist Methods

A write failure occurs when the value being stored in the bitcell is unable to be flipped. For example, to write the bitcell in Figure 4, the bitline (BL) is held high and BLB is held low. In order for the internal state to flip, pass-gate transistor XR must be able to pull node QB below the switching threshold of the left inverter. A ratioed fight is occurring between XR and PR, therefore transistor PR is usually made weak, to make writing easier. The downside to making the pull up transistor minimum sized is that it increases the VT variation of this transistor.

The goal of write assist methods is to further weaken the pull-up transistor or strengthen the pass-gate transistor. There are several ways to accomplish this. The first is to increase the pass-gate to pull-up ratio, however because we are operating in sub-threshold sizing is not an efficient knob. The second method is to collapse V_{DD}, which weakens the pull-up transistors. The third and fourth methods involve strengthening the pass-gate transistors by either boosting the WL V_{DD} or reducing the BL [7]. These methods strengthen the passgate by increasing its V_{GS}. The downside to boosting the WL V_{DD} is that it reduces half selected cell stability. The weakness of reducing the BL V_{SS} is that it increases the BL swing, which increases the total write energy.

4.3.2 Read Assist Methods

Read failures can occurs in two ways. The first is that the bitcell is flipped during a read operation (referred to as read failure). This occurs when the XL and NL1 transistors (Figure 4) are sinking the large amount of charge from the highly capacitive BL, and the Q node rises above the trip point of the right inverter. In order to increase read stability, the pull-down transistor is made stronger than the pass-gate. The second type of read failure occurs when the voltage difference between the BL and BLB is not large enough for the sense amp to determine the correct value (referred to as read access). This happens in sub-threshold especially due to the BL leakage current in unaccessed cells causing the BL voltage to droop. Because the I_{ON}/I_{OFF} ratio is reduced in sub-threshold, it is feasible for the leakage current through the unaccessed rows to pull the BL low at the same rate that the on current is pulling BLB low. This leakage current can be reduced by having less bitcells sharing the same bitline or by using one of the assist methods discussed below.

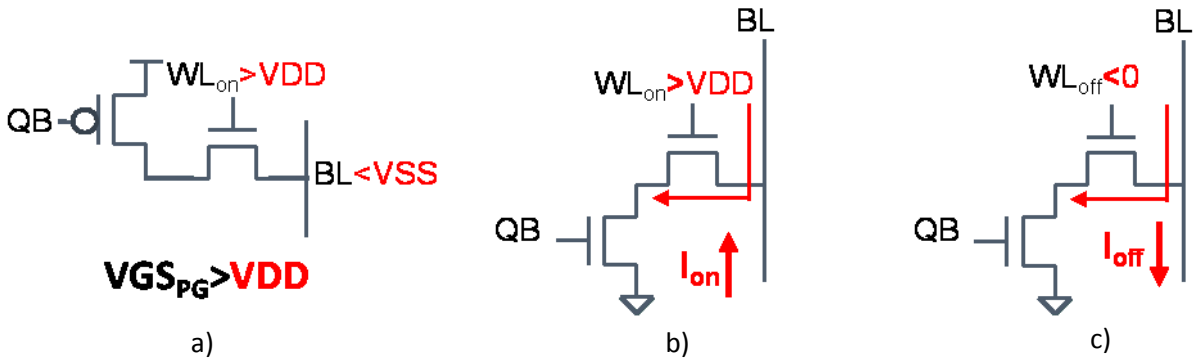

Figure 8. (a) increasing the pass-gate V_{GS} allows for easier writing of the bitcell; (b-c) boosting the on current and reducing off current improves read access.

There are two goals involved in read assist methods. The first is to improve the stability of the cross-coupled inverters during the read by either raising the bitcell V_{DD} or reducing its VSS [7]. While raising bitcell VDD has been shown by [7] to result in larger gains in RSNM, the advantage of reducing the bitcell VSS is that it significantly reduces read delay due to the body effect strengthening both the pull-down and pass-gate transistors. The second goal is improve read access by increasing the read current (I_{ON}) and reducing the BL leakage in unaccessed cells (I_{OFF}). The read current can be increased by boosting the WL V_{DD}. The downside here is that by strengthening the passgate, you reduce the stability of the cross-coupled inverters. In order to reduce bitline leakage current, the WL VSS is reduced to a negative voltage.

4.3.3 Bitcell Toplogies

The bitcell topologies under test include: traditional 6T, 8T [8], 10T Schmidt Trigger [9], and a new design featuring an 8T asymmetric Schmitt Trigger. This bitcell uses single-ended reading and asymmetric inverters, similar to the asymmetric 5T bitcell in to improve read margin. By using an asymmetrical design, the trip point of the ST inverter is increased, resulting in higher read stability. Because the 5T bitcell has only one access transistor, write assist methods must be used when trying to write a '1' into the bitcell. The advantage that this design has over the 5T bitcell is that it is written like a traditional 6T bitcell, which eliminates the need for write assist methods. The WL is pulsed high during both a read and write, and the WWL is only pulsed high during a write. In simulation, this bitcell achieves 86% higher RSNM than the 6T cell and 19% higher RSNM than the 10T ST bitcell with no VT variation added.

4.4 Evaluation Metrics

Each of the bitcells and assist method combinations will be evaluated on the percentage reduction of read and write V_{MIN} compared to the nominal case (6T bitcell with no assist methods).

4.5 Results

To compare bitcell topologies for subthreshold and to test assist features, a test chip was designed by a former student and fabricated in MITLL 180 nm FDSOI. This technology is specifically optimized for subthreshold operation by using an undoped channel to reduce capacitance and improve V_T control [28]. The optimizations result in a 50x reduction in energy-delay product compared to bulk silicon. The chip

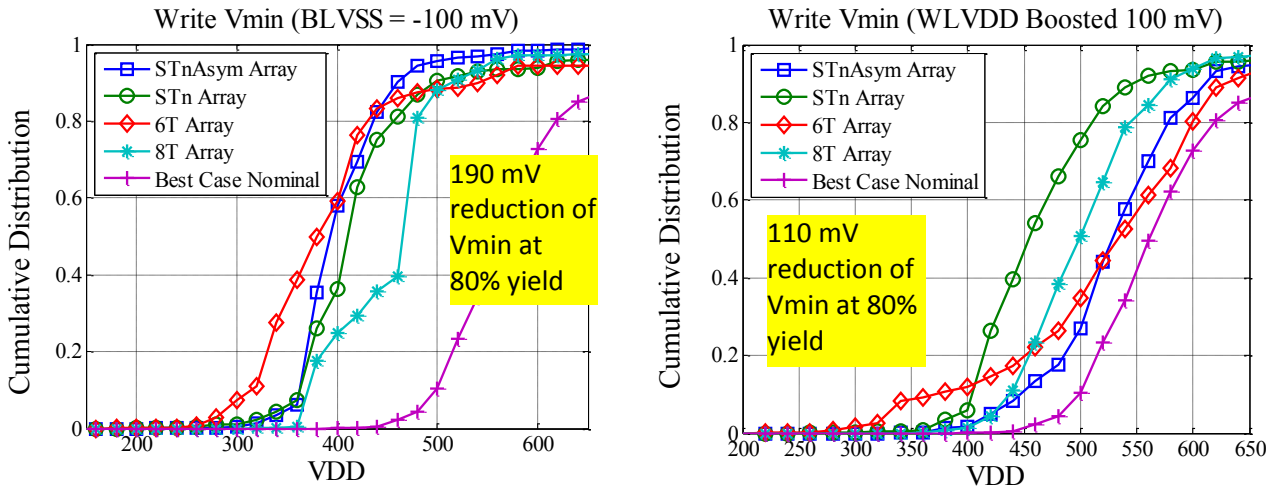

Figure 9. (left) effect of BL V_{SS} reduction on write V_{MIN}; (right) effect of WL V_{DD} boosting on write V_{MIN}; best case nominal refers to the bitcell with the lowest write V_{MIN} without the use of assist methods

contains four SRAM arrays, with each array containing two four-Kb banks. The banks' dimensions are 128 rows by two 16 bit words. The 6T and 8T cells are sized iso-area; the ST and asymmetric ST bitcells are also iso-area and suffer a 33% area penalty over the 6T and 8T bitcells. Because the main objective was reducing V_{MIN}, the chip was tested at 20 kHz to ensure that timing errors would not occur.

Because the test chip was fabricated during the first run of a new technology (MITLL 180nm FDSOI), the yield was not ideal. We found full columns to be non-functional as well as a relatively high number of random bit failures. However, even with the non-ideal yield we were able to obtain some interesting results. The first result was that the SRAM proved to be write limited, meaning that the write V_{MIN} exceeded the read V_{MIN}. The best case write V_{MIN} at 80% yield was 620 mV, and the best case read V_{MIN} was 440 mV at 80% yield. This number was chosen because the yield of some of the arrays even at nominal voltage was below 90%. Therefore in order to capture the trends of the various assist methods, we chose to use a yield value of 80% in order to negate the effect of these outliers. The 8T bitcell offered the lowest read V_{MIN} which is surprisingly only 10% lower than the other three bitcells. This is interesting because in simulation, the RSNM of the asymmetric ST and 10T ST bitcells was much higher than the 6T bitcell. What we observed was that there seems to be a discrepancy between the spice models and silicon data. This is most likely due to the technology being relatively immature during its first fabrication run. As a result, it was difficult to compare bitcell topologies, which ended up producing very similar results

Table 2. Percentage reduction in write V_{MIN} relative to write V_{MIN} without assist methods

Bitcell	BLV_{SS}	WLV_{DD}
6T	30%	3%
8T	23%	12%
10T ST	27%	18%
Asym. ST	30%	7%

in silicon.

Although bitcell measurements yielded inconclusive results, we can still evaluate assist features. The results from the different write assist methods are shown in Figure 9 and Table 2. Based on these figures, we conclude that BL V_{SS} reduction is the most effect method for reducing write V_{MIN}. This method outperforms the WL V_{DD} boost method across each of the bitcells. It is interesting to note that the 6T bitcell and Asymmetric ST bitcell achieve the lowest write V_{MIN} at 430 mV, a reduction of 190 mV compared to the best case without assist methods.

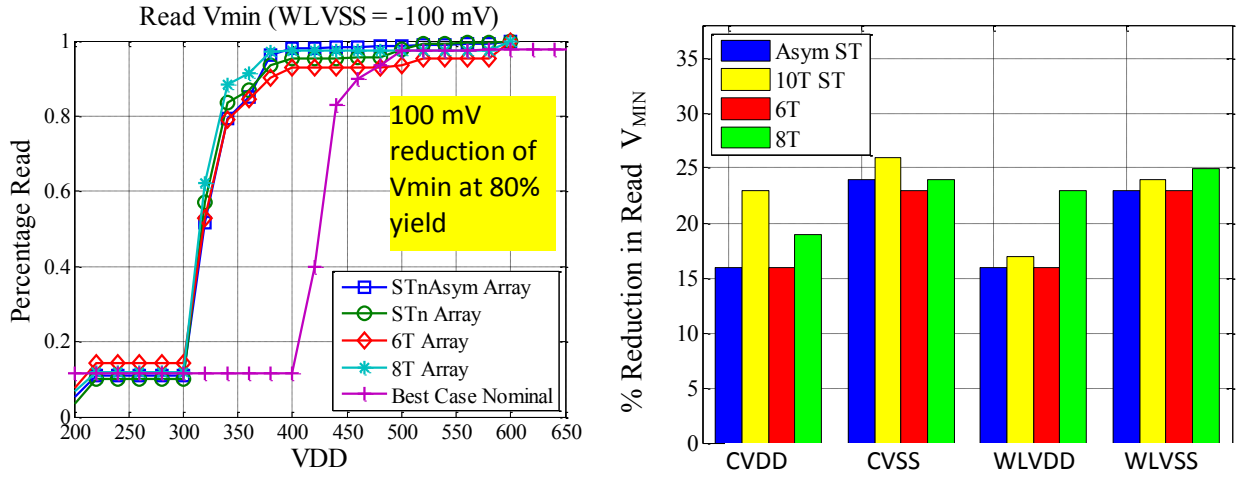

Figure 10. (left) effect of WL VSS reduction on read V_{MIN}; (right) comparison of read assist methods

As seen in Figure 10a, the WL VSS reduction resulted in a 100 mV reduction in read V_{MIN} for each of the bitcells. The interesting trend with this plot is that each of the bitcells had almost identical read V_{MIN} values. This would suggest using a combination of the 6T bitcell and WL VSS reduction is the most area efficient strategy for reducing read V_{MIN}. Based on the results from Figure 10b, reducing WL V_{SS} and bitcell V_{SS} consistently improved the read V_{MIN} for each of the bitcells. This suggests that bitline leakage was a major contributor to reduced read margin. It is also interesting to note that increasing the bitcell V_{DD} had the greatest impact on the 10T ST bitcell and WL V_{DD} boosting had the most positive effect on the 8T bitcell. Again, process features in the new technology most likely masked the effects of topological differences in the cells.

5. Virtual Prototyping (ViPro) Tool for Memory Subsystem Design Exploration and Optimization

5.1 Motivation

Increased variability, large arrays, and complexity increases make memory design a huge challenge for both conventional SRAM and emerging memory cell technologies. While process scaling has enabled ever-larger embedded memories, scaling issues such as device variability, leakage, soft error susceptibility, and interconnect delay make memory design increasingly difficult. As a result, how we will design efficient, robust SRAMs below the 32nm process technology node or how we will replace SRAM with emerging memory technologies remain largely open questions. Researchers have proposed promising circuit techniques, but they tend to address only individual components of the memory. However, a change in any one of the key memory circuits or in the core cell technology will alter the

optimal circuit topologies, partitioning, and architecture for the entire memory. For example, a larger new low-leakage bitcell could allow more cells on a bitline, so the net bit-density impact of the new cell becomes difficult to evaluate without a complete re-optimization of the memory circuits and architecture. We can no longer innovate in one portion of the memory while ignoring the effects our innovation could have on the overall memory and system design. Without the proper support structure and tools, it would be nearly impossible to re-design and re-optimize an entire memory by hand every time we try a new circuit, much less explore a technique's impact across different technologies and applications. Back-of-the-envelope estimation of overheads and impact on SRAM global metrics early in the design flow tends to be ad-hoc and dependent on assumptions that vary from designer to designer. Alternatively, implementing complete SRAM prototypes to evaluate each new technique impractically increases design time and reduces productivity. Thus, there is a need for a methodology through which designers can generate and evaluate prototypes at every step of the SRAM design process that account for process and circuit level issues in terms of global metrics.

5.2 Prior Art

There are a few memory design tools available, but they do not support integrated process-circuit-system co-design like ViPro. Architecture level modeling tools like CACTI [29] are used by computer architects to obtain quick estimates of SRAM access time, power, and area. CACTI 6.0 [30] facilitates high level design space exploration by using an optimization cost function that accounts for a user-weighted combination of delay, leakage, dynamic power, cycle time and area. ViPro also supports architectural exploration, but it differs from CACTI in two key ways. First, CACTI makes fixed assumptions regarding the circuits comprising the SRAM, so it optimizes at the architecture level only. ViPro allows designers to generate circuit information (via simulation) specific to any given technology or to add/alter the underlying circuits. Thus, it supports circuit-architecture co-design, which leads to

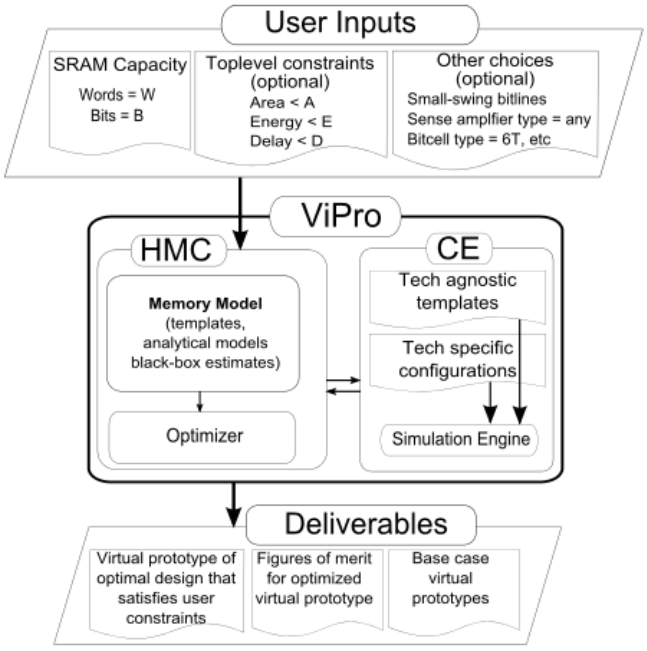

Figure 11. Top-level structure of ViPro. The characterization engine (CE) and hierarchical meta-compiler (HMC) model implement the two main features of ViPro- technology-agnosticism and a flexible hierarchical memory model

better overall designs. Second, CACTI supports a limited set of technologies and assumes ITRS parameters for its calculations. These assumptions may not be accurate, especially for advanced processes. ViPro uses a technology-agnostic simulation environment (TASE) [32] to characterize its circuit components in any process using SPICE simulations before generating the virtual prototypes, so it uses accurate technology-specific circuit parameters for any process.

ViPro was originally developed at UVA [31]. In order to evaluate different designs, the tool works in two phases. The first phase called TASE [32] (Technology Agnostic Simulation Environment) combines process information with templates for common simulations to create parameterized characterizations of memory components in any given process technology with SPICE level accuracy. The second phase uses a hierarchical model of the memory array to optimize the design for a given set of constraints. By using a hierarchical model, we allow for the tool to be easily extensible and scalable, which is important because the SRAM design space is constantly changing and evolving. Each component in the SRAM is included in the model, allowing for accurate computation of the global figures of merit. A key feature of the tool is that different blocks in the hierarchical model can take on different degrees of accuracy; some blocks can use extremely high level estimates of behavior (e.g. energy = constant, delay = constant) while other blocks can use detailed models or full SPICE netlists. This allows a designer to experiment with different options and to receive rapid estimates of macro level metrics. The current version of the tool allows for brute optimization (using energy and delay as the metrics) of a single bank SRAM design.

5.2 Hypothesis

By extending the existing ViPro tool to support multi-bank designs, 8T bitcell designs, read and write assist methods, yield evaluation, and a circuit and architectural level co-optimization engine we will be able to explore a much larger design space and run a much larger set of novel experiments.

5.3 Approach

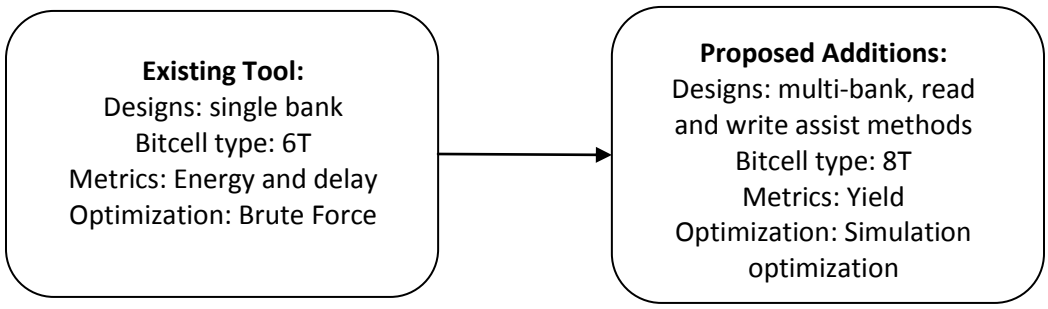

Figure 12. Chart showing the existing features of ViPro and the proposed additions

5.3.1 Expanding the Design Space

The first step in expanding the design space exploration that ViPro is capable of performing is adding support for multi-bank designs. Most large SRAM arrays are broken into banks because there are a limited number of cells that can be placed on the same bitline. By supporting multi-bank design, the tool will be able to evaluate much larger capacity arrays (i.e. > 100 KB), which are common in today's SoCs. In addition to evaluating multi-bank designs, we are also proposing to support designs

which use the 8T bitcell (Figure 4). This bitcell is common in level one cache due to its dual port design. It also introduces new design challenges due to its single ended read structure. Finally, we are proposing to support designs which use read and write assist methods to improve the robustness of SRAMs in the presence of variability. Assist methods introduce new tradeoffs between energy, speed, area and yield which are difficult to evaluate because they are dependent on many factors such as technology node, bitcell architecture, and design constraints. Therefore it is important to be able to evaluate the tradeoffs between the various methods under different system constraints.

5.3.2 Yield Evaluation

Because memories can contain millions of cells, it is not feasible to run standard Monte Carlo simulations in order to calculate yield. Therefore we propose to use the methodology outlined in section 3 for evaluating write failure probabilities. This methodology offers a two order of magnitude speed up over importance sampling, at a relatively low cost in error. In order to evaluate read access failure probabilities, we propose to incorporate the statistical model outlined in [33] to the tool. The advantage of this model is that it takes into account that the probability of the worst case bitcell being paired with the worst case sense amp is very low. This allows for more accurate approximations of yield. In addition, this model takes into account the effect of architectural features on yield, such as the number of bits per column and the number of columns per sense amp. Because sense amps must be pitch matched to the bitcells (to reduce area and increase regularity), increasing the number of words per row (or level of column muxing) reduces the total number of sense amps (and therefore reduces the offset of the worst case sense amp). In addition, more column muxing allows for the transistors in the SA circuit to be upsized, thus reducing variation. The trade off is that extra column muxing increases delay. This tradeoff is just one experiment that the tool will be able to evaluate.

5.3.3 Simulation Optimization

Currently the tool supports optimization through a brute force search. This means that every possible combination of knobs is simulated in order to determine the best case energy or delay point. While this method works for small design spaces, as the number of optimization knobs expands, this method will no longer be feasible. A more suitable approach is for the optimization engine to learn from the previous iterations, and make educated guesses as to which combination of knobs will result in a more optimal design. This form of optimization is known as simulation optimization. By using simulation optimization, we will be able to reduce the total number of iterations required to reach the optimal design point, based on the criteria set by the designer.

5.4 Evaluation Metrics

Because of ViPro's unique design and functionality, it is difficult to make a direct comparison to previous tools such as CACTI. Therefore, the tool will be evaluated based on the novel contributions and experiments that it will enable. The optimization engine will be evaluated based on the speedup gained over brute force optimization.

5.5 Goals and Anticipated Contributions

The major goal of this chapter is to expand the capabilities of the existing ViPro tool to allow it to perform circuit and architectural co-optimization of a much larger design space. Because the use

of assist methods is a relatively new idea, the ability to evaluate how the tradeoffs in yield, energy and delay change across technology node, operating voltage, memory size and memory architecture is a valuable asset to today's memory designers. For example, in memories with high bitline leakage, using a negative WL V_{SS} might be more beneficial than using a boosted WL for increasing read access reliability. The ability to perform these types of experiments is what makes the tool highly impactful. Expanding the tool to support multi-bank designs also makes the tool more valuable because most of today's large cache designs require this type of architecture. In addition, because reliability is such an issue with large capacity nanoscale memories, it is important to understand how circuit and architectural level design decisions affect yield. This feature could lead to new design strategies for increasing yields in nanoscale SRAMs.

6. Canary-Based PVT Tracking System for Reducing Write V_{MIN}

6.1 Motivation

As discussed throughout this paper, reducing SRAM V_{MIN} to gain quadratic energy savings is one of the largest challenges in SRAM design today. One of the major reasons for this is process, voltage, and temperature (PVT) variation. For commercial designs, it is important to be able to guarantee functionality across a wide range of PVT corners. Traditional methods of guard-banding consider the worst case scenario for setting the operating voltage at design time. This conservative approach ensures reliable operation across the worst PVT corners; however it also sacrifices potential energy savings because the full range of V_{MIN} is large when accounting for the worst case [34]. Because the circuit is not always operating in the worst case PVT corner, there is a potential to regain some of this lost energy. One alternative approach is to use a closed loop feedback system to track PVT variations. Using this method, the operating voltage could be optimally set in real time based on outputs from the tracking system.

6.2 Prior Art

The canary based feedback system was first introduced in [34] as a method for reducing the standby voltage in a 90 nm SRAM. Each bitcell has a data retention voltage (DRV) which is the minimum voltage that a cell can maintain its data. Local variation sets the sigma of this distribution, and global effects tend to shift the mean [34]. Because a small set of canary cells cannot replicate the statistics of the entire array, the canaries can only track global variation, not local variation [34]. By tracking global PVT variation, the canary cells can effectively remove the need to guard-band for these global conditions. The canary cells are designed specifically to fail at higher voltages than the average core cell. This is achieved in [34] by using a header to modulate the virtual V_{DD} of the canary cells. In order to detect failures, the internal nodes of the canary cells are wired directly to control logic through a buffer. The canary array contains multiple sets of cells tuned specifically to fail in regular intervals at voltages higher than the DRV of the core cells (Figure 13) [34]. Using multiple failure thresholds in the canary array allows for a direct tradeoff between reliability and power.

The closed loop controller lowers the standby voltage until a failure is detected in the canary cells. Each set of canary failures corresponds to a failure probability in the core array, which is determined through simulation. The control loop is tuned to ensure that the voltage of the core array never drops below the array wide DRV [34]. However in some applications where bit failures aren't as costly, the control loop can be tuned to allow for more aggressive scaling at the cost of likely bit failures in the core

array. This method was shown by [34] to offer a 30x power savings over traditional guard banding techniques with an area overhead of only 0.6%.

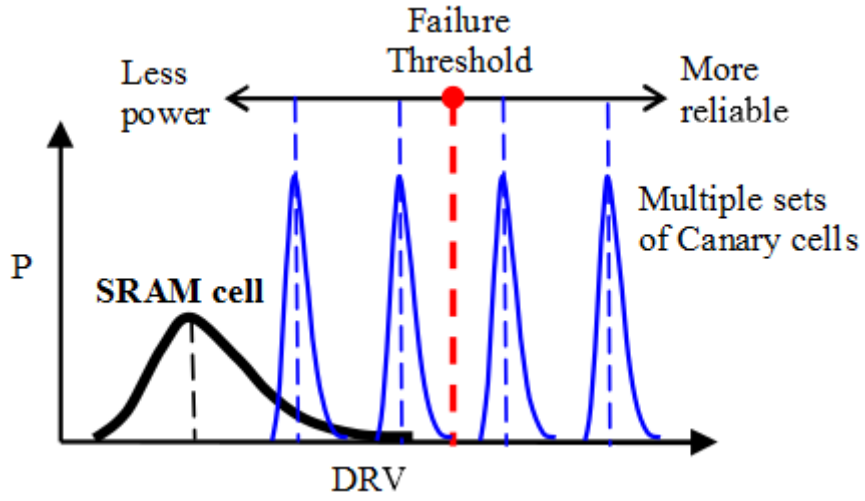

Figure 13. Canary cells are tuned to fail in regular intervals at voltages higher than the DRV of the core cells [34]

6.3 Hypothesis

We hypothesize that a similar canary based closed loop feedback system can be implemented to increase the power savings over traditional guard banding. As a proof of concept, we will look specifically at implementing this system for reducing write V_{MIN}. While a full canary system would need to monitor PVT variation in both the read and write path, we have chosen in this case to limit our scope to only the write operation in this work.

6.4 Approach

We propose a closed loop canary based feedback system for optimally setting V_{DD} during the SRAM write operation. First, the minimum operating voltage of the core array must be determined through simulation. This distribution can be rapidly obtaining using the importance sampling method described in [23]. There are two potential methods for tuning the canary failure thresholds. The first is to use a reverse assist method such as WL droop or BL V_{SS} boost in order to shift the mean of the distribution. In this case, it is important that the word line pulse width of the canary cells is equal to that of the core array. The second method is to decrease the length of the word line pulse width of the canary cells. Based on our results from Chapter 3, we know that a shorter WL pulse width will result in a lower average write V_{MIN}. These two methods will be evaluated in terms of area overhead, ease of implementation, and effectiveness in tracking global PVT variations. In order to detect write failures, the internal nodes of the canary cells can be wired directly out to logic as in [34]. Finally, a control loop will be implemented to monitor failures within the canary banks and adjust the write voltage as close to the V_{MIN} of the core array as possible.

6.5 Evaluation Metrics

The system will be evaluated in terms of total energy savings over conventional guard banding approaches and total area overhead.

6.6 Anticipated Contributions

The major goal of this chapter is to develop a closed loop canary based system to track global PVT variations, set the write voltage to the optimal level, and provide energy savings over conventional guard banding approaches. The results of this project could provide a method for further reducing SRAM V_{MIN} in nanoscale designs without reducing reliability.

7. Research Tasks

Table 3 outlines the tasks, status and relevant publications of each research goal.

Table 3. Research tasks and timeline

Subject	#	Task description	Status/Target	Publications
BSN Memory Design	1	Rev 1: Define Specifications/Design	Completed	
	2	Rev 1: Simulation/Verification	Completed	
	3	Rev 1: Layout	Completed	
	4	Rev 1: Chip Testing	Completed	[JMB1][JMB3][JMB1][JMB2]
	5	Rev 2: Define Specifications/Design	Completed	
	6	Rev 2: Simulation/Verification	Completed	
	7	Rev 2: Layout	Completed	
	8	Rev 2: Chip Testing	August-2013	[JMB7][JMB7]
Dynamic Write V_{MIN} Estimation	1	Survey existing techniques for determining VMIN	Completed	
	2	Create a new model for estimating dynamic VMIN	Completed	
	3	Verify Model Accuracy	Completed	[JMB4][JMB4]
Sub-threshold bitcell analysis	1	Test Chip	Completed	
Sub-threshold assist method analysis	1	Test Chip	Completed	[JMB2][JMB2]
Virtual Prototyping Tool	1	Expand existing capabilities to support multi-bank design	Completed	[JMB6][JMB6]
	2	Add support for 8T bitcell	Completed	[JMB5]
	3	Verify Model Accuracy	June-2013	
	4	Integrate read and write assist features	September-2013	
	5	Integrate yield estimation	December-2013	[JMB8][JMB8]

	6	Optimize using simulation optimization algorithm	August-2014	[JMB9][JMB9]
	7	Integrate yield estimation into optimization algorithm	November-2014	
Canary Feedback System	1	Evaluation of canary design	December 2013	
	2	Design of voltage control loop	February 2014	
	3	Simulation/Verification	March 2014	
	4	Layout	May 2014	
	5	Chip Testing	October 2014	[JMB11]
Write up	1	Thesis Writing	January-2015	

8. Publications

8.1 Current

[JMB1] F. Zhang, Y. Zhang, J. Silver, Y. Shakhsheer, M. Nagaraju, A. Klinefelter, J. Pandey, **J. Boley**, E. Carlson, A. Shrivastava, B. Otis, and B. H. Calhoun, "A Battery-less 19μW MICS/ISM-Band Energy Harvesting Body Area Sensor Node SoC," *ISSCC*, February 2012.

[JMB2] **J. Boley**, J. Wang, and B. H. Calhoun, "Analyzing Sub-Threshold Bitcell Topologies and the Effects of Assist Methods on SRAM V_{MIN}", *JLPEA*, April 2012.

[JMB3] Y. Zhang, F. Zhang, Y. Shakhsheer, J. Silver, A. Klinefelter, M. Nagaraju, **J. Boley**, J. N. Pandey, A. Shrivastava, E. J. Carlson, A. Wood, B. H. Calhoun, and B. Otis, "A Batteryless 19 μW MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications," *JSSC,* 2013.

[JMB4] **J. Boley**, V. Chandra, R. Aitken, and B. Calhoun, "Leveraging Sensitivity Analysis for Fast, Accurate Estimation of SRAM Dynamic Write V_{MIN}," *DATE*, 2013.

8.2 Anticipated

[JMB5] J. Boley, P. Beshay, and B. Calhoun, "Virtual Prototyping (ViPro) Tool for Memory Subsystem Design Exploration and Optimization," TECHCON, 2013

[JMB6] P. Beshay, J. Boley, and B. Calhoun, "SRAM Optimization using Simulated Annealing"

[JMB7] A. Banerjee, J. Boley, and B. Calhoun, "Subthreshold SRAM Design Featuring Low Energy Read operation"

[JMB8] Evaluation of SRAM Assist Methods on Top Level Design Metrics

[JMB9] Optimization of SRAMs for Improved Yield

[JMB10] Using simulation optimization for SRAM design space exploration

[JMB11] Canary based closed-loop control system for optimizing write V_{DD}

References

[1] A. Bhavnagarwala, X. Tang, and J. Meindl "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *JSSC*, pp. 658-665, 2001.

[2] Seevinck, E.; List, F.J.; Lohstroh, J. Static-noise margin analysis of MOS SRAM cells. *IEEE J. Solid-State Circuits* **1987**, *22*, 748-754.

[3] J. Ryan, S. Khanna, and B. Calhoun, "An analytical model for performance yield of nanoscale SRAM accounting for the sense amplifier strobe signal," *ISLPED*, 2011.

[4] G. Z. Yang*, Body Sensor Networks*. London, U.K.: Springer-Verlag 2006.

[5] E. Carlson, K. Strunz, and B. Otis "A 20 mV Input Boost Converter With Efficient Digital Control for Thermoelectric Energy Harvesting," *JSSC*, Vol. 45, No. 4, April 2010.

[6] Wang, A.; Chandrakasan, A.; Kosonocky, S. Optimal Supply and Threshold Scaling for Sub-threshold CMOS Circuits. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI 2002*, Pittsburgh, PA, USA, 25–26 April 2002; pp. 7-11.

[7] Mann, R.W.; Nalam, S.; Wang, J.; Calhoun, B.H. Limits of bias based assist methods in nano-scale 6T SRAM. In *Proceedings of the 11th International Symposium on Quality Electronic Design*, San Jose, CA, USA, 22-24 March 2010; pp. 1-8.

[8] Verma, N. Chandrakasan, A.P. A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy. *IEEE J. Solid-State Circuits* **2008***, 43*, 141-149.

[9] Kulkarni, J.P.; Kim, K.; Roy, K. A 160 mV Robust Schmitt Trigger Based Subthreshold SRAM," *IEEE J. Solid-State Circuits* **2007**, *42*, 2303-2313.

[10] J. Wang, A. Singhee, R. Rutenbar, and B. Calhoun, "Statistical modeling for the minimum standby supply voltage of a full sram array," *ESSCIRC,* pp. 400-403, 2007.

[11] J. Wang and B. Calhoun, "Minimum supply voltage and yield estimation for large srams under parametric variations," *IEEE Transactions of VLSI Systems,* pp. 2120-2125, 2011.

[12] D. Khalil, M. Khellah, N. Kim, Y. Ismail, T. Karnik, and V. De, "Accurate Estimation of sram dynamic stability," *IEEE Transactions of VLSI Sytems*, pp. 1639-1647, 2008.

[13] M. Sharifkhani and M. Sachdev, "SRAM cell stability: A dynamic perspective," *JSSC,* vol. 44, pp. 609-619, 2009.

[14] W. Dong, L. Peng, and G.M. Huang, "SRAM dynamic stability: theory, variability and analysis," *ICCAD*, pp. 378-385, 2008.

[15] J. Wang, S. Nalam, and B.H. Calhoun "Analyzing static and dynamic write margin for nanometer SRAMs," *ISLPED*, pp. 129-134, 2008.

[16] S. Nalam, V. Chandra, R. Aitken, and B.H. Calhoun, "Dynamic write limited minimum operating voltage for nanoscale SRAMs," *DATE*, pp. 1-6, 2011.

[17] S.O. Toh, Z. Guo, and B. Nikolic, "Dynamic SRAM stability characterization in 45nm CMOS," *IEEE Symposium on VLSI Circuits,* pp. 35-36, 2010.

[18] M. Yamaoka, K. Osada, and T. Kawahara, "A cell-activation-time controlled SRAM for low-voltage operation in DVFS SoCs using dynamic stability analysis," *ESSCIRC,* pp. 286-289, 2008.

[19] A. Bhavnagarwala et al., "Fluctuation limits and scaling opportunites for cmos sram cells," *IEDM*, PP. 659-662, 2005.

[20] B. Zhang, A. Arapostathis, S. Nassif, and M. Orshansky, "Analytical modeling of sram dynamic stability," *ICCAD*, pp. 315-322, 2006.

[21] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanosclaed cmos," *IEEE Transactions on Computer –Aided Design of Integrated Circuits and Systems*, pp.1859-1880, 2005.

[22] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," *DAC,* pp. 69-72, 2006.

[23] T. Doorn, E. Maten, J. Croon, A. Bucchianico, and O. Wittich, "Importance sampling monte carlo simulations for accurate estimation of sram yield," *ESSCIRC,* pp. 230-233, 2008.

[24] A. Singhee and R. Rutenbar, “Statistical blockade: a novel method for very fast monte carlo simulation of rare circuit events, and its application,” *DATE, 2007.*

[25] A. Singhee, J. Wang, B. Calhoun, and R. Rutenbar, “Recursive statistical blockade: an enhanced technique for rare even simulation with application to sram circuit design,” *VLSID,* pp. 131-136, 2008.

[26] Z. Guo, et al., “Large-scale read/write margin measurement in 45 nm CMOS SRAM arrays,” *Proc. Symp. VLSI Circuits,* pp.42-43, 2008.

[27] Y. Tsukamoto, et al., “Worst-case analysis to obtain stable read/write dc margin of high density 6t-sram-array with local vth variability,” *ICCAD,* pp. 398-405, 2005.

[28] Vitale, S.A.; Wyatt, P.W.; Checka, N.; Kedzierski, J.; Keast, C.L. FDSOI process technology for subthreshold-operation ultralow-power electronics. In *Proceedings of the IEEE* **2010**, *98*, 333-342.

[29] P. Shivakumar and N. P. Jouppi, “Cacti 3.0: An integrated cache timing, power, and area model,” West.Res.Lab.,Tech. Rep., 2002.

[30] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, “Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0,” in *MICRO* 40, pp. 3–14, 2007.

[31] Nalam, S., M. Bhargava, K. Mai, and B. H. Calhoun, “Virtual prototype (ViPro): An early design space explorationand optimization tool for SRAM designers,” *DAC*, 2010.

[32] Nalam, S., M. Bhargava, K. Ringgenberg, K. Mai, and B. H. Calhoun, "A Technology-Agnostic Simulation Environment (TASE) for Iterative Custom IC Design across Processes", *ICCD*, pp. 523-528, 2009.

[33] M. H. Abu-Rahma, K. Chowdhury, J. Wang, Z. Chen, S. Yoon, M. Anis, “A methodology for statistical estimation of read access yield in SRAMs,” *DAC*, 2008.

[34] J. Wang and B. Calhoun, “Canary replica feedback for Near-DRV standby VDD scaling in a 90 nm SRAM,” *CICC*, 2007.